GETWD

Carefully manage buffer sizes (getwd() is deprecated)

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5998 bytes

Attack Category	• Malicious Input
Vulnerability Category	• Buffer Management • Buffer Overflow
Software Context	• File Path Management
Location	• unistd.h
Description	The getcwd() function copies an absolute pathname of the current working directory to the array pointed to by buf, which is of length size. If the current absolute path name would require a buffer longer than size elements, NULL is returned, and errno is set to ERANGE; an application should check for this error, and allocate a larger buffer if necessary. If buf is NULL, the behavior of getcwd() is undefined. As an extension to the POSIX.1 standard, Linux (libc4, libc5, glibc) getcwd() allocates the buffer dynamically using malloc() if buf is NULL on call. In this case, the allocated buffer has the length size unless size is zero, when buf is allocated as big as necessary. It is possible (and, indeed, advisable) to free() the buffers if they have been obtained this way. get_current_dir_name, which is only prototyped if _GNU_SOUdepRCE is defined, will malloc(3) an array big enough to hold the current directory name. If the environment variable PWD is set, and its value is correct, then that value will be returned. getwd, which is only prototyped if _BSD_SOURCE or _XOPEN_SOURCE_EXTENDED is defined, will not malloc(3) any memory. The buf argument should be a pointer to an array at least PATH_MAX bytes long. getwd does only return the first PATH_MAX bytes of the actual pathname. Note that PATH_MAX need not be a compile-time constant;

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	it may depend on the filesystem and may even be unlimited. The getwd() function shall determine an absolute pathname of the current working directory of the calling process and copy a string containing that pathname into the array pointed to by the path_name argument. If the length of the pathname of the current working directory is greater than ({PATH_MAX}+1) including the NULL byte, getwd() shall fail and return a null pointer. For portability and security reasons, use of getwd is deprecated.
APIs	**Function Name** | **Comments** getwd | getcwd |
Method of Attack	Since the user cannot specify the length of the buffer passed to getwd(), use of this function is discouraged. The length of a pathname described in {PATH_MAX} is file system-dependent and may vary from one mount point to another, or might even be unlimited. It is possible to overflow this buffer in such a way as to cause applications to fail or possible system security violations.
Exception Criteria	N/A
Solutions	**Solution Applicability** | **Solution Description** | **Solution Efficacy** Should always be applied based on deprecation of getwd. | Replace getwd with getcwd |
Signature Details	char *getcwd(char *buf, size_t size); char *get_current_dir_name(void); char *getwd(char *buf);
Examples of Incorrect Code	`/* This is a simple, normal example of getwd */` `/* The behavior of getwd if dir >1024 (as noted in the description) is not portable */` `/* but it is not malloc based therefore, even when NULL is returned for the "too large" */` `/* scenario, this is clearly not a robust-enough function */` `{` `char dir[1024], *s;`

	`` s = getwd (dir); if (s == 0) { printf ("Error getting pwd: %s \n", dir); return 1; } printf ("Current directory is %s \n", dir); return 0; } ```
Examples of Corrected Code	``` char cwd[PATH_MAX+1]; { if (getcwd(cwd, PATH_MAX+1) == NULL) { perror("getcwd failed"); } else { return 0; } } ```
Source References	• man getwd • http://www.annodex.net/cgi-bin/man/man2html?getwd+3 • http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/dangers-c.html[3] • http://www.opengroup.org/onlinepubs/009695399/functions/getwd.html
Recommended Resources	
Discriminant Set	**Operating System**: • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com